TP noté (sujet A) : Nombres parfaits

Votre nom : Code Capytale : **47a1-2303954**

Partie A.I : questions indépendantes

1. Écrire une fonction `somme` qui prend en argument une liste L et retourne la somme des éléments de L. On ne pourra pas utiliser la fonction "`sum`".
2. Écrire une fonction `carre` qui prend en argument une liste L et retourne une liste de même taille qui contient les carrés de chaque valeur de L.
3. Écrire une fonction `inversion` qui à une liste L retourne cette même liste mais dans l'ordre inverse.
4. Écrire une fonction `factorielle` qui à un entier n retourne la valeur de $n!$.

Partie A.II : nombres parfaits

5. Écrire une fonction `listeDiviseurs` qui à un entier $n \geq 1$ retourne la liste des diviseurs (positifs) de n.

Soit $n \in \mathbb{N}^*$ un entier. On dit que $d \in \mathbb{N}$ est un diviseur strict de n si d divise n et $d \neq n$. Par exemple, les diviseurs stricts de 9 sont 1 et 3, tandis que 1 n'a pas de diviseur strict. On dit que n est un nombre parfait si la somme des diviseurs stricts de n est égale à n. Par exemple, 6 est parfait car ses diviseurs stricts sont 1, 2 et 3 et leur somme fait 6. Par convention, 1 n'est pas parfait.

6. Écrire une fonction `estParfait` qui prend en argument un entier $n \in \mathbb{N}^*$ et retourne `True` si n est parfait, `False` sinon.
7. Écrire une fonction `listeParfait` qui prend en argument un entier $m \in \mathbb{N}^*$ et qui retourne la liste des m premiers nombres parfaits. *Ne pas tester pour $m \geq 5$!*

Pour tout $n \in \mathbb{N}^*$, on définit la n-ième somme harmonique lacunaire, notée ici S_n, comme étant la somme des inverses de tous les diviseurs de n (y compris n lui-même). Par exemple

$$S_4 = \frac{1}{1} + \frac{1}{2} + \frac{1}{4} = \frac{7}{4} \qquad S_6 = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{6} = 2$$

On conjecture la propriété suivante : pour tout $n \in \mathbb{N}^*$, l'entier n est un nombre parfait si et seulement si $S_n = 2$.

8. Écrire une fonction `conjParfait` qui prend en argument un entier $N \geq 2$ et qui teste la conjecture ci-dessus pour tous les nombres n allant de 1 à N, mais ne retourne rien. Pour chaque entier n :
   - Si la conjecture est vraie, on affichera dans la console "conjecture vraie pour n = " puis le nombre.
   - Si la conjecture est fausse, on affichera dans la console "conjecture FAUSSE pour n = " puis le nombre.

   La conjecture vous semble-t-elle vraie ? (Répondre dans un commentaire).

Si vous avez fini TOUTES les questions, demandez le sujet B et faites la partie B.II, puis III.
N'utilisez pas le code du sujet B. Continuez à la suite du code que vous avez déjà écrit.

Partie III : bonus (nécessite d'avoir fait la partie B.II !)

- Bien avant Mersenne, Euclide a conjecturé que M_n est premier si et seulement si $1 + 2 + 3 + \cdots + M_n$ est un nombre parfait. Écrire une fonction `conjEuclide` qui prend en argument un entier $N \geq 2$ et qui teste la conjecture d'Euclide pour tous les entiers n allant de 2 à N, mais ne retourne rien. Comme pour la question 7, on affichera le résultat dans la console pour chaque entier n. La conjecture vous semble-t-elle vraie ? *Ne pas tester pour $N \geq 13$!*
- (Bon courage...) Écrire une fonction `sousChaine` qui à une chaine de caractères S et un entier k retourne la longueur de la plus grande sous-chaine de S composée d'au plus k caractères différents. Par exemple `sousChaine("abcba",2)` retournera 3, obtenu avec la sous-chaine `"bcb"` (et non `"abba"` : il faut prendre des caractères consécutifs).

TP noté (sujet B) : nombres de Mersenne

Votre nom : Code Capytale : **7b07-2304034**

Partie B.I : questions indépendantes

1. Écrire une fonction `comptage` qui prend en argument une liste L et un élément a et qui retourne le nombre d'occurrences de a dans L. On ne pourra pas utiliser la méthode `L.count()`.
2. Écrire une fonction `pair` qui prend en argument une liste d'entiers L et retourne `True` si L contient un entier pair, `False` sinon.
3. Écrire une fonction `unSurDeux` qui à une liste L retourne la liste constituée des éléments de L ayant un indice pair.
4. Écrire une fonction `triangle` qui prend en arguments trois nombres positifs a, b, c et qui retourne `True` si on peut construire un triangle dont les côtés ont pour longueurs a, b, c, et `False` sinon. *Indication : cela équivaut à dire qu'un des côtés n'est pas plus grand que la somme des deux autres.*

Partie B.II : nombres de Mersenne

On rappelle qu'un nombre $n \geq 2$ est premier si ses seuls diviseurs (positifs) sont 1 et n. Par convention, 1 n'est pas premier.

5. Écrire une fonction `estPremier` qui à un entier $n \geq 2$ renvoie `True` si n est premier, `False` sinon.

Soit $n \in \mathbb{N}^*$ un entier. On appelle <u>n-ième nombre de Mersenne</u> l'entier $M_n := 2^n - 1$. De plus, on dit que n est <u>Mersenne-premier</u> si M_n est premier. Par exemple, 3 est Mersenne-premier car $M_3 = 2^3 - 1 = 7$ est premier. Par contre, 4 n'est pas Mersenne-premier car $M_4 = 2^4 - 1 = 15$ n'est pas premier.

6. Écrire une fonction `listeMersenne` qui prend en argument un entier $m \in \mathbb{N}^*$ et qui retourne la liste des m premiers nombres qui sont Mersenne-premiers. *Ne pas tester pour $m \geq 8$!*

Mersenne a conjecturé que pour tout $n \in \mathbb{N}^*$, l'entier n est premier si et seulement si M_n est premier.

7. Écrire une fonction `conjMersenne` qui prend en argument un entier $N \geq 2$ et qui teste la conjecture ci-dessus pour tous les entiers n allant de 1 à N, mais ne retourne rien. Pour chaque entier n :
   - Si la conjecture est vraie, on affichera dans la console “conjecture vraie pour n = “ puis le nombre.
   - Si la conjecture est fausse, on affichera dans la console “conjecture FAUSSE pour n = “ puis le nombre.

   La conjecture vous semble-t-elle vraie ? Répondre dans un commentaire.

Si vous avez fini TOUTES les questions, demandez le sujet A et faites la partie A.II, puis la partie III. N'utilisez pas le code du sujet A. Continuez à la suite du code que vous avez déjà écrit.

Partie III : bonus (nécessite d'avoir fait la partie A.II !)

- Bien avant Mersenne, Euclide a conjecturé que M_n est premier si et seulement si $1 + 2 + 3 + \cdots + M_n$ est un nombre parfait. Écrire une fonction `conjEuclide` qui prend en argument un entier $N \geq 2$ et qui teste la conjecture d'Euclide pour tous les entiers n allant de 2 à N, mais ne retourne rien. Comme pour la question 7, on affichera le résultat dans la console pour chaque entier n. La conjecture vous semble-t-elle vraie ? *Ne pas tester pour $N \geq 13$!*
- (Bon courage...) Écrire une fonction `sousChaine` qui à une chaine de caractères S et un entier k retourne la longueur de la plus grande sous-chaine de S composée d'au plus k caractères différents. Par exemple `sousChaine("abcba",2)` retournera 3, obtenu avec la sous-chaine `"bcb"` (et non `"abba"` : il faut prendre des caractères consécutifs).